PyWPS 4 Development restart

Jáchym Čepický¹

1http://les-ejk.cz

Geoinformatics, Prague 2013



TOC

1 PyWPS



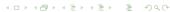
- OGC WPS on the Server
- Since 2006
- Python
- http://pywps.wald.intevation.org
- http://github.org/geopython/pywps



PyWPS - what it is NOT

- PyWPS is no analytical tool or engine. It does not perform any type of geospatial calculation.
- PyWPS is not special XML parser or generator. It does not validate your GMLs against given schemas (yet), it does not build GML from Python objects.
- It is not complicated





PyWPS - what it is NOT

- PyWPS is no analytical tool or engine. It does not perform any type of geospatial calculation.
- PyWPS is not special XML parser or generator. It does not validate your GMLs against given schemas (yet), it does not build GML from Python objects.
- It is not complicated





PyWPS - what it is NOT

- PyWPS is no analytical tool or engine. It does not perform any type of geospatial calculation.
- PyWPS is not special XML parser or generator. It does not validate your GMLs against given schemas (yet), it does not build GML from Python objects.
- It is not complicated







rather bike, then a car



#small





#modular





#fast



#easy





#slick





#accessories (GRASS, GDAL, Shapely, #python) PyWPS

- Started from scratch
- Use Python 2.7 (for future 3.0 migration)
- Try different interpreters of Python (pypy)
- Easy parsing with lxml
- Prepare for next WPS version
- Change of the whole process concept

- Started from scratch
- Use Python 2.7 (for future 3.0 migration)
- Try different interpreters of Python (pypy)
- Easy parsing with lxml
- Prepare for next WPS version
- Change of the whole process concept



- Started from scratch
- Use Python 2.7 (for future 3.0 migration)
- Try different interpreters of Python (pypy)
- Easy parsing with lxml
- Prepare for next WPS version
- Change of the whole process concept



- Started from scratch
- Use Python 2.7 (for future 3.0 migration)
- Try different interpreters of Python (pypy)
- Easy parsing with lxml
- Prepare for next WPS version
- Change of the whole process concept



- Started from scratch
- Use Python 2.7 (for future 3.0 migration)
- Try different interpreters of Python (pypy)
- Easy parsing with lxml
- Prepare for next WPS version
- Change of the whole process concept



- Started from scratch
- Use Python 2.7 (for future 3.0 migration)
- Try different interpreters of Python (pypy)
- Easy parsing with lxml
- Prepare for next WPS version
- Change of the whole process concept



- Started from scratch
- Use Python 2.7 (for future 3.0 migration)
- Try different interpreters of Python (pypy)
- Easy parsing with lxml
- Prepare for next WPS version
- Change of the whole process concept





#geopython 2006



#geopython 2013

- lxml http://lxml.org
- GRASS-WPS, GRASS-Python
- Werkzeug http://werkzeug.pocoo.org/
- Python 3
- Django
- ...



Process definition in PyWPS 3

```
class Buffer (WPSProcess):
    def init (self):
        WPSProcess. init (self, identifier="buffe
        self.addComplexInput(identifier="input", ...
        self.addLiteralOutpu(identifier="input", ...
    def execute(self):
        return
```

Process definition in PyWPS 4 (proposal)

- 2-level APIs
- Providing the low-level API (integration with GRASS, writing a compatibility api for PyWPS-3, and defining a bunch of processes programatically if the user needs that.
- The process is just a callable
- Inspired by Django
- Upper-level API for easy user-written processes

PyWPS 4 - upper-level

```
class Buffer(pywps.Process):
    class Input:
        distance = pywps.Literal(value_type=pywps.F
        layer = pywps.Complex(as_file=True, max_occ
    class Output:
        feature_count = pywps.Literal(value_type=pyr
        fortune = pywps.Literal(value_type
    def execute(self):
```

PyWPS 4 - low-level

```
make buffer(request):
make buffer process = pywps.create_process(
    identifier='makebuffer', title =...
    inputs=[
        pywps. Literal (identifier='distance', value
        pywps.BoundingBox(identifier='clip'),
    outputs =[
        pywps.Literal(identifier='feature count', v
        pywps. Literal (identifier='fortune', value ty
    handler=make buffer)
make buffer process()
```

- http://github.org/jachym/pywps-4
- Tests, Basic request parsing, process definition, . . .
- Alex Morega @mgax, Jorge de Jesus, Jachym Cepicky
- http://lists.osgeo.org/cgi-bin/mailman/ listinfo/pywps-dev



- http://github.org/jachym/pywps-4
- Tests, Basic request parsing, process definition, ...
- Alex Morega @mgax, Jorge de Jesus, Jachym Cepicky
- http://lists.osgeo.org/cgi-bin/mailman/listinfo/pywps-dev



- http://github.org/jachym/pywps-4
- Tests, Basic request parsing, process definition, ...
- Alex Morega @mgax, Jorge de Jesus, Jachym Cepicky
- http://lists.osgeo.org/cgi-bin/mailman/ listinfo/pywps-dev



- http://github.org/jachym/pywps-4
- Tests, Basic request parsing, process definition, . . .
- Alex Morega @mgax, Jorge de Jesus, Jachym Cepicky
- http://lists.osgeo.org/cgi-bin/mailman/ listinfo/pywps-dev



thank you! jachym@les-ejk.cz

http://github.org/jachym/pywps-4

questions?

